

---

# Formal verification and decision procedures in mathematics

Jeremy Avigad

Department of Philosophy

Carnegie Mellon University

<http://www.andrew.cmu.edu/~avigad>

# On the formalizability of mathematics

---

The development of mathematics towards greater precision has led, as is well known, to the formalization of large tracts of it, so that one can prove any theorem using nothing but a few mechanical rules. The most comprehensive formal systems that have been set up hitherto are the system of *Principia Mathematica (PM)* on the one hand and the Zermelo-Fraenkel axiom system of set theory (further developed by J. von Neumann) on the other. These two systems are so comprehensive that in them all methods of proof today used in mathematics are formalized, that is, reduced to a few axioms of rules and inferences. One might therefore conjecture that these axioms and rules of inference are sufficient to decide *any* mathematical question that can at all be formally expressed in these systems. It will be shown below that this is not the case...

(Kurt Gödel, *On formally undecidable propositions of Principia Mathematica and related systems I*, 1931)

# From principal to practice

---

For most of the twentieth century, Gödel's claim that most of mathematics can be formalized in *PM* and set theory was understood to be true “in principal.”

In practice, spelling out every inference in terms of axioms and rules of logic is tedious and difficult, even for very simple proofs.

Towards the end of the century, mechanized proof assistants were developed, however, which began to make formalization possible in practice.

(For historical references, see Freek Wiedijk's web page.)

# How they work

---

Ordinary textbook proofs do not look like formal derivations:

“...the first law may be proved by induction on  $n$ .”

“...by successive applications of the definition, the associative law, the induction assumption, and the definition again.”

“By choice of  $m$ ,  $P(k)$  will be true for all  $k < m$ .”

“Hence, by the well-ordering postulate...”

“From this formula it is clear that...”

“This reduction can be repeated on  $b$  and  $r_1$ ...”

“This can be done by expressing the successive remainders  $r_i$  in terms of  $a$  and  $b$ ...”

“By the definition of a prime...”

“On multiplying through by  $b$ ...”

“...by the second induction principle, we can assume  $P(b)$  and  $P(c)$  to be true...”

“Continue this process until no primes are left on one side of the resulting equation...”

“Collecting these occurrences, ...”

“...Theorem 10 allows us to conclude ...”

# How they work

---

In a proof system, declaring a theorem is tantamount to announcing a goal, i.e. the goal of proving it. Instructions are then used to reduce the goal to (hopefully) simpler ones.

For example, the logical “ $\wedge$ -introduction” rule reduces a goal of the form

$$X_1, X_2, \dots, X_n \Rightarrow Y \wedge Z$$

to the two subgoals

$$X_1, X_2, \dots, X_n \Rightarrow Y$$

$$X_1, X_2, \dots, X_n \Rightarrow Z.$$

The logical “ $\wedge$ -elimination” rule reduces a goal of the form

$$X_1, X_2, \dots, X_n, Y \wedge Z \Rightarrow W$$

to the subgoal

$$X_1, X_2, \dots, X_n, Y, Z \Rightarrow W.$$

# How they work

---

Here is an example of a proof script in Isabelle:

```
theorem (a::int) dvd b ==> a^n dvd b^n  
  apply (induct-tac n)  
  apply (subst power-0)+  
  apply (rule zdvd-1-left)  
  apply (subst power-Suc)+  
  apply (rule zdvd-zmult-mono)  
  apply (assumption)+  
  done
```

# How they work

---

**proof** —

**assume**  $a \text{ dvd } b$

**show**  $a^n \text{ dvd } b^n$

**proof** (*induct n*)

**show**  $a^0 \text{ dvd } b^0$

**proof** —

**have**  $a^0 = 1$

**by** (*rule power-0*)

**moreover have** ( $1 \text{ dvd } b^0$ )

**by** (*rule zdvd-1-left*)

**ultimately show** *?thesis*

**by** *simp*

**qed**

**next fix**  $n$

**assume**  $a^n \text{ dvd } b^n$

**show**  $a^{Suc\ n} \text{ dvd } b^{Suc\ n}$

**proof** —

**from prems have**  $a * a^n \text{ dvd } b * b^n$

**by** (*intro zdvd-zmult-mono*)

**moreover have**  $a^{Suc\ n} = a * a^n$

**by** (*rule power-Suc*)

**moreover have**  $b^{Suc\ n} = b * b^n$

**by** (*rule power-Suc*)

**ultimately show** *?thesis*

**by** *simp*

**qed**

# How they work

---

In this case, there is a one-line proof:

```
theorem (a::int) dvd b ==> a^n dvd b^n  
by (induct n, auto intro: zdvd-zmult-mono)
```

In words: “Use induction on  $n$ . The verification is straightforward, using theorem *zdvd-zmult-mono*.”

# Milestones

---

Recently there have been some milestone formalizations:

- Prime Number Theorem (Avigad et al., September 2004, using Isabelle)
- The four color theorem (Gonthier, December 2004, using Coq)
- The Jordan curve theorem (Hales, February 2005, using HOL light)

Hales has launched an ambitious effort to formally verify his proof of the Kepler conjecture (Google “Flyspeck”).

# The prime number theorem

---

Let  $\pi(x)$  denote the number of primes less than or equal to  $x$ .

The prime number theorem:  $\pi(x)/x$  is asymptotic to  $1/\ln x$ , i.e.

$$\lim_{x \rightarrow \infty} \pi(x) \ln x / x = 1.$$

Conjectured by Gauss and Legendre, on the basis of computation, around 1800; proved by Hadamard and de la Vallée Poussin in 1896.

Kevin Donnelly, David Gray, Paul Raff, and I used Isabelle to verify:

$$(\lambda x. \text{pi } x * \ln (\text{real } x) / (\text{real } x)) \text{ --- } > 1$$

# Why is formal verification interesting

---

*Overblown claim:* Mathematics is error-ridden and in danger of collapse.

For the most part, mathematics does fine. But errors are ubiquitous. Some are easily repaired; others are more substantial.

*Overblown claim:* The only thing that matters is that proofs are correct.

There are many facets to mathematical understanding. But correctness is important part of it.

*Overblown claim:* Formal verification is the most important role for computers in mathematical discovery.

Actually, from a mathematical perspective, it's one of the least interesting.

*Overblown claim:* Mathematicians should start verifying proofs today.

When the effort required to verifying a proof is commensurate with the required to write it up and referee it, verification will be worthwhile.

---

# Why formal verification is interesting

---

Formal verification is, in and of itself, a fascinating business, from many perspectives.

Computer science:

- Interesting issues in automated reasoning.
- Interesting database issues.
- Interesting interface issues.

Mathematics:

- Better theory is needed to support automated reasoning.
- Need a more robust modeling of mathematical proofs.
- Internal mathematical developments are needed to support verification.

# Why formal verification is interesting

---

## Philosophy:

- Need more robust study of mathematical language
- Need more robust study of mathematical concepts and their role in structuring theories
- Need more robust study of methods or reasoning and understanding

Tools and insights that are developed to support formal verification of mathematical theorems can also be relevant to other aspects of automated reasoning and verification.

In particular, formal verification of hardware and software systems is very important.

# Decision procedures and heuristic procedures

---

We would like to have procedures that are capable of verifying that an inference is correct.

Ideally:

- We would like to have decision procedures.
- They should be efficient.
- They should be general.
- They should construct axiomatic proofs of verified claims.

However:

- First-order logic is undecidable.
- Arithmetic is undecidable.
- Set theory is undecidable.

# Decision procedures and heuristic procedures

---

We can still look for:

- Search procedures that generally work well “in practice.”
- Domain-specific decision procedures.
- Domain-specific search procedures that work well in specific instances.

Let  $T$  be a formal axiomatic theory. In some cases,  $T$ , or an interesting fragment of  $T$ , is decidable.

Let  $\mathcal{M}$  be a mathematical structure, such as

$$\langle \mathbb{R}, 0, 1, +, \times, < \rangle$$

Sometimes the theory of  $\mathcal{M}$  is decidable.

## Two important examples

---

Let  $\mathcal{N}$  be the structure  $\langle \mathbb{N}, 0, 1, +, < \rangle$ .

**Theorem (Presburger, 1929)** The theory of  $\mathcal{N}$ , a.k.a. *Presburger arithmetic*, is decidable.

In fact, decision procedures are implemented in Isabelle, Coq, ...

Let  $\mathcal{M}$  be the structure  $\langle \mathbb{R}, 0, 1, +, \times, < \rangle$ .

**Theorem (Tarski, around 1930).** The theory of  $\mathcal{M}$ , a.k.a. the theory of *real-closed fields*, is decidable.

Sean McLaughlin and John Harrison have recently implemented a proof-producing version in HOL light.

**Corollary** Euclidean geometry is decidable!

# Real closed fields

---

But the story doesn't end here.

- RCF procedures are slow (and arguably misguided, for the types of inferences that come up in ordinary proofs).
- Worse: they do not extend to straightforward inferences with monotone functions, trigonometric functions, exponentiation and logarithm, etc.

Problem: nontrivial parts of mathematics are undecidable. Two options:

- Use full decision procedures in more restricted settings.
- Use “heuristic procedures” in more general settings.

Harvey Friedman and I have developed one approach to verifying inequalities that come up in “ordinary” proofs.

# Inequalities between real-valued expressions

---

The algorithmic challenge: determine the validity of boolean combinations of equalities and inequalities between real-valued expressions.

An example:

$$0 < x < y \rightarrow (1 + x^2)/(2 + y)^{17} < (1 + y^2)/(2 + x)^{10}$$

Another example:

$$0 < x < y \rightarrow (1 + x^2)/(2 + e^y) < (2 + y^2)/(1 + e^x)$$

# Inequalities between real-valued expressions

---

A slightly more complicated example: verify

$$\left(1 + \frac{\varepsilon}{3(C + 3)}\right) \cdot n < Kx$$

using the following hypotheses:

$$n \leq (K/2)x$$

$$0 < C$$

$$0 < \varepsilon < 1$$

Idea: work forwards and backwards, applying obvious monotonicity rules.

# Heuristic procedures for the reals

---

## Problems:

1. Case splits: e.g.  $st > 0 \equiv (s > 0 \wedge t > 0) \vee (s < 0 \wedge t < 0)$ .
2. Nondeterminism: e.g. many ways to show  $s + t < u + v + w$ .

## Observations:

1. “Straightforward” inferences usually don’t need case splits.
2. In practice, Fourier-Motzkin is efficient for linear inequalities.
3. Modulo cases over signs, the same thing works for the multiplicative fragment of the reals.

Idea: find a principled way of combining the “local” decision procedures.

# The Fourier-Motzkin procedure

---

**Theorem.** The theory of  $\langle \mathbb{R}, 0, 1, +, < \rangle$  has quantifier-elimination, and so is decidable.

**Proof.** It suffices to show that if  $\varphi$  is quantifier-free,  $\exists x \varphi$  is equivalent to a quantifier-free formula.

Note:

- Can put  $\varphi$  in disjunctive normal form.
- $\exists x (\theta \vee \eta)$  is equivalent to  $\exists x \theta \vee \exists x \eta$ .
- $s \neq t$  is equivalent to  $s < t \vee t < s$ .
- $s \not\leq t$  is equivalent to  $t < s \vee s = t$ .

So, it suffices to assume  $\varphi$  is a conjunction of equalities and strict inequalities.

# The Fourier-Motzkin procedure

---

Expressions that don't involve  $x$  can be brought outside the existential quantifier.

Using rational coefficients, can put expressions involving  $x$  in *pivot form*:

- $x = s$
- $x < s$
- $s < x$

$\varphi$  is a conjunction of these.

If any conjunct has the form  $x = s$ ,  $\exists x \varphi(x)$  is equivalent to  $\varphi(s)$ , and we're done.

# The Fourier-Motzkin procedure

---

Otherwise,  $\varphi$  is a conjunction of formulas of the form  $s_i < x$  and  $x < t_j$ .

It is not hard to check that  $\exists x \varphi$  is equivalent to

$$\bigwedge_{i,j} s_i < t_j.$$

Notes:

- Can allow multiplicative coefficients from any computable field.
- The procedure requires double-exponential time in principle, but works quite well in practice.

## Other “local” decision procedures

---

**Theorem.** The theory of  $\langle \mathbb{R}, 0, 1, \times, \sqrt[n]{\cdot}, < \rangle$  also has quantifier elimination. In fact, one can allow constants in any computable subfield of  $\mathbb{R}$ .

The idea:

- $\langle \mathbb{R}^{>0}, 1, \times, < \rangle$  looks just like  $\langle \mathbb{R}, 0, +, < \rangle$
- $\exists x \varphi(x)$  is equivalent to  $\exists x > 0 \varphi(x) \vee \varphi(0) \vee \exists x < 0 \varphi(-x)$ .
- If  $s$  and  $t$  are positive,  $-s < t$  is true,  $s < -t$  is false, and  $-s < -t$  is equivalent to  $t < s$ .

Exercise:

**Theorem.** The theory of  $\langle \mathbb{R}, 0, 1, \exp, \ln, < \rangle$  is decidable.

Can we combine these?

# Combining decision procedures

---

**Theorem.** Suppose  $T_1$  and  $T_2$  are “locally finite” and decidable. Suppose that the languages are disjoint, except for the equality symbol. Then the universal fragment of  $T_1 \cup T_2$  is decidable.

In particular, if  $T_1$  and  $T_2$  have only infinite models, they are locally finite.

This allows you to design decision procedures for individual theories and then put them together!

With additional hypotheses on the source theories, the decision procedures can be made efficient (Nelson-Oppen, Shostak, ...).

# Combining decision procedures

---

First idea: one can “separate variables” in universal formulas.

That is,  $\forall \vec{x} \varphi(\vec{x})$  is equivalent to  $\forall \vec{y} (\varphi_1(\vec{y}) \vee \varphi_2(\vec{y}))$ , where  $\varphi_1$  is in the language of  $T_1$ , and  $\varphi_2$  is in the language of  $T_2$ .

To do this, just introduce new variables to name subterms.

Second idea: the Craig interpolation theorem.

**Theorem.** Suppose  $\psi_1$  is a sentence in  $L_1$  and  $\psi_2$  is a sentence in  $L_2$ , such that  $\vdash \psi_1 \rightarrow \psi_2$ . Then there is a sentence  $\theta$  in  $L_1 \cap L_2$  such that

- $\vdash \psi_1 \rightarrow \theta$
- $\vdash \theta \rightarrow \psi_2$

# Combining decision procedures

---

Let  $\varphi$  be any universal sentence, equivalent to  $\forall \vec{x} (\varphi_1(\vec{x}) \vee \varphi_2(\vec{x}))$ .

Then  $T_1 \cup T_2 \vdash \varphi$  if and only if there is  $\theta$  in the common language, such that

- $T_1 \cup \{\neg\varphi_1(\vec{x})\} \vdash \theta(\vec{x})$
- $T_2 \cup \{\neg\varphi_2(\vec{x})\} \vdash \neg\theta(\vec{x})$

We can assume  $\theta$  is in disjunctive normal form. All that each disjunct can do is declare certain variables equal to one another, and others unequal!

Use the decision procedures for  $T_1$  and  $T_2$  to test each possibility.

# Theories of real inequalities

---

Good news: the results extend to the case at hand.

Let  $T_1$  be the theory of  $\langle \mathbb{R}, 0, 1, +, -, < \rangle$ .  $T_1$  is decidable.

Let  $T_2$  be the theory of  $\langle \mathbb{R}, 0, 1, \times, \div, \sqrt[n]{\cdot}, < \rangle$ .  $T_2$  is decidable.

Let  $T = T_1 \cup T_2$ . By Nelson-Oppen methods, the universal fragment of  $T$  is decidable.

Bad news:  $T$  is too weak; it doesn't prove  $2 \times 2 = 4$ .

# Heuristic procedures for the reals

---

A better version: let  $f_a(x) = ax$  for rational constants  $a$ .

Let  $T_{add}[\mathbb{Q}]$  be the theory of  $\langle \mathbb{R}, 0, 1, +, -, <, \dots, f_a, \dots \rangle$ .

Let  $T_{mult}[\mathbb{Q}]$  be the theory of  $\langle \mathbb{R}, 0, 1, \times, \div, \sqrt[n]{\cdot}, <, \dots, f_a, \dots \rangle$ .

Let  $T_{common}[\mathbb{Q}] = T_{add}[\mathbb{Q}] \cap T_{mult}[\mathbb{Q}]$ .

Let  $T[\mathbb{Q}] = T_{add}[\mathbb{Q}] \cup T_{mult}[\mathbb{Q}]$ . This theory seems to be very useful.

$T_{add}[\mathbb{Q}]$ ,  $T_{mult}[\mathbb{Q}]$ ,  $T_{common}[\mathbb{Q}]$  all have quantifier elimination. But Nelson-Oppen methods fail when there is a nontrivial overlap.

The situation here is much more complex!

# Theories of real inequalities

---

Think of  $T[\mathbb{Q}]$  as:

- real-closed fields without distributivity (except for constants)
- a shotgun wedding of the additive and multiplicative theories.

It seems to cover very many “obvious” calculations.

**Theorem.** Let  $f(x_1, \dots, x_k)$  be a polynomial over  $\mathbb{Q}$ . Then  $f$  is nonzero on  $[0, 1]^k$  if and only if  $T[\mathbb{Q}]$  proves that fact.

This provides a lower bound on the strength of  $T[\mathbb{Q}]$  on universal assertions. For an upper bound:

**Theorem.**  $T[\mathbb{Q}]$  proves  $\forall x (x^2 - 2x + 1 \geq \varepsilon)$  if and only if  $\varepsilon < 0$ .

In fact, the size of a minimal interpolant depends on  $\varepsilon$ .

# Theories of real inequalities

---

Here are some of our results.

- $T[\mathbb{Q}]$  has good normal forms.
- Valid equations are independent of the ordering.
- $T[\mathbb{Q}]$  is undecidable.
- In fact, the  $\forall\forall\forall\exists\dots\exists$  fragment is complete r.e.
- Assuming that the solvability of Diophantine equations in the rationals is undecidable, then so is the existential fragment of  $T[\mathbb{Q}]$ .

Most important:

- The universal fragment of  $T[\mathbb{Q}]$  is decidable.

More generally, we consider theories  $T[F]$ , for arbitrary computable subfields  $F$  of  $\mathbb{R}$ .

# The universal fragment of $T[F]$

---

Let  $\varphi \equiv \forall \vec{x} (\varphi_{add}(\vec{x}) \vee \varphi_{mult}(\vec{x}))$ .

The language of  $T_{common}[F]$  has atomic formulas  $x_i = ax_j$ ,  $x_i < ax_j$ .  
(We can assume each  $x_i > 0$ , and  $x_1 = 1$ .)

**Theorem.**  $T[F]$  proves  $\varphi$  iff there is a quantifier-free interpolant  $\theta(\vec{x})$  in the language of  $T_{common}[F]$  such that

- $T_{add}[F] \cup \{\neg\varphi_{add}(\vec{x})\} \vdash \theta(\vec{x})$
- $T_{mult}[F] \cup \{\neg\varphi_{mult}(\vec{x})\} \vdash \neg\theta(\vec{x})$ .

Notes:

- Can take  $\theta$  e.g. to be a conjunction of disjuncts of the form  $x_i < ax_j$  and  $x_i \leq ax_j$ .
- Can use this and “transfer” to show that  $T[\mathbb{R}]$  is a conservative extension of  $T[\mathbb{A}]$ .

# The universal fragment of $T[F]$

---

Let  $\varphi \equiv \forall \vec{x} (\varphi_{add}(\vec{x}) \vee \varphi_{mult}(\vec{x}))$ .

**Theorem.** The following are equivalent:

1.  $T[F]$  doesn't prove  $\varphi$ .
2. The union of  $T_{add}[F] \cup \{\neg\varphi_{add}(\vec{x})\}$  and  $T_{mult}[F] \cup \{\neg\varphi_{mult}(\vec{x})\}$  is consistent.
3. There is a complete type  $\Gamma(\vec{x})$  in the language of  $T_{common}[F]$  such that
  - $T_{add}[F] \cup \{\neg\varphi_{add}(\vec{x})\} \cup \Gamma(\vec{x})$  and
  - $T_{mult}[F] \cup \{\neg\varphi_{mult}(\vec{x})\} \cup \Gamma(\vec{x})$are both consistent.

# The universal fragment of $T[F]$

---

$T[F] \vdash \varphi$  iff for every complete type  $\Gamma(\vec{x})$  in the language of  $T_{common}[F]$ , there is a finite subset  $\Gamma'(\vec{x})$  such that either

$$\forall \vec{x} \left( \bigwedge \Gamma'(\vec{x}) \rightarrow \varphi_{add}(\vec{x}) \right) \quad \text{or} \quad \forall \vec{x} \left( \bigwedge \Gamma'(\vec{x}) \rightarrow \varphi_{mult}(\vec{x}) \right)$$

holds in the reals.

One can characterize all the complete types,  $\Gamma(\vec{x})$ , in terms of what they say about pairs  $\{x_i, x_j\}$ .

With work, the assertion above can be expressed by a restricted class of formulas in the language of real closed fields, with a predicate for  $F$ .

With more work, one can show that this class is decidable (assuming  $F$  is computable and  $F \cap \mathbb{A}$  is decidable).

# Undecidability results

---

**Theorem.** There is a model of  $T[F]$  where the solutions to the equation

$$x(1 + x) = x + x^2$$

are exactly the  $x \in F$ .

**Corollary.** An existential sentence  $\varphi$  over  $F$  if and only if in any model of  $T[F]$ ,  $\varphi$  has witnesses among the  $a$  with  $a(1 + a) = a + a^2$ .

**Corollary.** If Diophantine equations in the rationals are unsolvable, then so is the set of existential consequences of  $T[\mathbb{Q}]$ .

# Undecidability results

---

**Theorem.** There is a model of  $T[F]$  and elements  $\mu, \kappa, \lambda$  such that solutions  $x \in [1, \mu]$  to

$$(\kappa + x)(\lambda + x) = \kappa\lambda + \kappa x + \lambda x + x^2$$

are exactly the positive integers.

**Corollary.** Let  $\varphi$  be a Diophantine equation over the positive integers. Then  $\varphi$  has a solution in the positive integers if and only for every model  $\mathcal{M}$  of  $T[F]$ , and every  $\mu, \kappa, \lambda \in \mathcal{M}$ , if

$$\{x \in [1, \mu] \mid (\kappa + x)(\lambda + x) = \kappa\lambda + \kappa x + \lambda x + x^2\}$$

contains 1 and is closed under  $+1$ , then  $\varphi$  has solutions in that set.

**Corollary.** The set of  $\forall\forall\exists\dots\exists$  consequence of  $T[F]$  is complete r.e.

# Heuristic procedures for the reals

---

Our decidability results are not practical. But the proofs provide ideas and guidelines.

General strategy for amalgamation:

- Maintain a database of facts in the common language.
- Iteratively use each of  $T_{add}[\mathbb{Q}]$  and  $T_{mult}[\mathbb{Q}]$  to add new facts.

Issues:

- Heuristically, how to decide *which* facts to focus on?
- When to split on cases?
- How to look for disjunctions?
- How to incorporate distributivity?
- How to amalgamate other local decision or heuristic procedures?

# Conclusions

---

Formally verified mathematics is becoming increasingly important:

- Proofs are getting very complex.
- Proofs rely on extensive computations.

New approaches are needed:

- Interesting fragments of mathematics are undecidable.
- Heuristic procedures are brittle, hard to extend, and unpredictable.

What we need are *principled* search procedures:

- Build heuristics on sound theory.
- Develop more useful classifications of mathematical contexts.

# Conclusions

---

Continued progress will require

- thoughtful reflection
- good theory
- solid engineering

This makes the field an auspicious combination of theory and practice.